**АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА**

**ПРЕДСТАВЯНЕ НА ЧИСЛО**

1. **Наивно решение**

Наивното решение е да се образуват всевъзможните суми от последователни, цели числа и да се проверява, коя от тях е равна на *N*. Дори и при това решение могат да се съобразят някои неща – че сумирането трябва да продължава докато сумата е по-малка или равна от *N* и че първото (най-малко) число от сумата не може да бъде по-голямо от *N/2.*

Такова решение формално е със сложност *O(N2)*, но с много малка константа и, при ръчно измерване, би трябвало да решава задачата за първите четири тестови примера и да получи 40 точки.

Реализацията на решението е в **nsum\_slow.cpp**.

Ако някой не е съобразил „рязането“, което беше описано по-горе, а направи съвсем пълно изчерпване на сумите, то би трябвало да получи точки само на първия пример.

1. **Решение със сложност O( )**

Това решение се основава на следните съображения:

Нека *N=a+(a+1)+(a+2)+…..+(a+r-1).*

Тогава *N=r\*a+1+2+….+(r-1)=r\*a+(r-1)\*r/2* или получаваме *2\*N* *= r\*(2\*a+r-1).*

Както се вижда, *r* е броя на последователните числа в представянето на *N* и лесно се съобразява, че *r* и *2\*a+r-1* са с различна четност.Освен това е ясно, че, при *a>0,* е изпълнено *2\*a+r-1>r.* От тук пък следва, че *r\*r<2\*N.*

Това ни дава идеята да проверяваме всички числа *r* от *1* до и което от тях е делител на *2\*N* и самото *r* или *(2\*N)/r* е нечетно, то това определя едно представяне на *N* като сума от последователни числа.

Тъй като нечетният делител на *2\*N* всъщност трябва да е делител на *N*, то можем да проверяваме числата *x* от 1 до , като за всяко от тях гледаме дали *x* е делител и *x,* *N/x* или и двете са нечетни. Тъй като проверяваме само до , то, ако и *x* и *N/x* са нечетни и са различни, то трябва да преброяваме две представяния на *N* като сума от последователни числа.

Това решение ще получи 100 точки и е реализирано в **nsum\_fast.cpp**.

Възможни са и междинни решения, които ще получат точки между 40 и 100.

*Автор: Руско Шиков*